Patent Claims

1. A method for actuating function units in a processor, where a configuration phase involves a series of primary instruction words which comes from a translation of a program code being divided into a series of instruction word parts, with a program cycle involving instruction words which actuate the processor being constructed in the full instruction word length to form a VLIW and being buffer-stored in an instruction word memory (cache), characterized in that
a first step involves a primary instruction word being divided, in the configuration phase, into the series of a particular number of instruction word parts which are used for constructing a respective VLIW during the execution phase, with a respective first and second FIW (Function Instruction Word Part) (4), (7) being preceded with an associated first or second operating code (2), (5) which thus determines how the cache's memory location taken up by the respective FIW is handled in the execution phase,
in that the respective first or second operating code (2), (5) is followed by an associated first or second tag (3), (6) which represents the information regarding which first or second FU (19), (20) actuates the respective FIW,
in that the first or second operating code (2), (5) and its associated first or second tag (3), (6) are combined with the respective first and second FIWs (4), (7) to form the first and second TVLIW containers (11), (12), all of these representing the TVLIW (1),
in that a second step involves the respective available TVLIW (1) being converted into an HVLIW (10) in the configuration phase, with the HVLIW (10) containing a preceding general header (13),

in that the HVLIW (10) with its code-compressed structure replaces all functions of the TVLIW (1).

2. The method as claimed in claim 1, characterized in that a "Command Code" mode of operation of the HVLIW (10) and its associated general header (13) are implemented,
in that the general header (13) is followed directly by the first and second FIWs (4), (7) required for constructing the VLIW (22),
in that this general header (13) stores the information, in coded form, which indicates all combinations regarding which first and second FIW (instruction word part) (4), (7) is provided, after decoding, in the execution phase, for actuating a respective first and/or second FU (function unit) (19), (20) in the processor (21),
in that the general header (13) stores which first and/or second FIW (4), (7) takes up memory locations in the cache (26) and whether or which operations are to be executed with the respective memory content in the execution phase in the cache (26) when constructing the VLIW (22).

3. The method as claimed in claims 1 and 2, characterized in that
the first part of the general header (13) is provided with a header mode (14) which contains information about the "Command Code" mode of operation of the HVLIW (10) and of the general header (13),
in that this is followed by a second part which stores, coded as table values, the respective most needed combination regarding which of the respective FUs is actuated by which first or second FIW (3), (7),
in that a third part is connected as CE information (16) and contains a pointer which

refers to a provided location in the dictionary (9),
in that the last part of the general header (13) which is provided is the supplementary information (17).

4. The method as claimed in claim 1, characterized in that a "reference instruction" mode of operation of the HVLIW (10) and of the general header (13) contained is implemented in which the FIWs provided for constructing the VLIW (22) in the execution phase are buffer-stored in the cache (26), the associated header mode (14) bearing a correspondingly decodeable tag for this "reference instruction" mode of operation,
in that the "reference instruction" mode of operation is initiated by a specific HVLIW (10) which contains an address statement which is used to refer to a reference instruction,
in that the subsequent HVLIW (10), which likewise bears the tag for the "reference instruction" mode of operation, contains a relative address for the address statement provided by the reference,
in that this has a mask appended to it for the FUs which are to be excluded from the actuation.

5. The method as claimed in claim 3, characterized in that the address statement of the specific HVLIW (10) which initiates the "reference instruction" mode of operation refers to the preceding HVLIW (10).

6. The method as claimed in claim 3, characterized in that the address statement of the specific HVLIW (10) which initiates the "reference instruction" mode of operation refers to a general address.

7. The method as claimed in claims 1 to 3, characterized
in that the execution phase involves the HVLIW (10) being decoded in a decoder1 (23) which is equipped with a header decoder (24), a CMDT (25), a cache (26) and a cache miss repair logic unit (27), the HVLIW (10) being buffer-stored in the cache (26), and
the header decoder (24) identifies the mode of operation of the general header (13) from the header mode (14) stored therein,
in that the identified header mode (14) is taken as a basis for decompressing the values of the FU-C information (14) which are provided in the general header (13) by means of a comparison with the CMDT (25) and in conjunction with the CE information (16) which is likewise taken from the general header (13),
in that the identified header mode (14) is taken as a basis for processing the supplementary information (17) in the general header (13),
in that possible incorrect access during buffer-storage in the cache (26) (cache miss) is remedied by the execution of an error handling routine in the cache miss repair logic unit (28),
in that the valid VLIW (22) is provided at the output of the decoder1 (23).